

Producing Smooth Motion Compensated Frames by Combining Multiple Interpolation Results

Inventor

Christophe Souchard

Background

Field of Invention

[0001] The present invention relates generally to video frame interpolation, and more specifically to combining results of multiple motion compensated interpolations in order to produce smooth motion in interpolated frames.

Background of Invention

[0002] Motion compensation frame interpolation is an important research field, with many applications such as video compression, video format conversation, special effects production, etc. Prior art frame interpolation relies on motion estimation, and therefore its performance is data dependant. Although certain methods of motion estimation work acceptably well with certain data conditions, no single prior art technique for motion estimation performs appropriately for all data types under all conditions. Different values for an interpolated frame between two existing frames can be computed, each of which may be more or less appropriate for the specific case.

[0003] What is needed are methods, systems and computer program products that generate multiple motion compensated interpolated frames, and robustly fuse the results

into a single, output interpolated frame of a higher quality than could be achieved by any single technique alone.

Summary of Invention

[0004] An interpolation manager constructs multiple motion compensated interpolated frames between two existing frames, and then fuses the multiple motion compensated interpolated frames into a single output frame. In some embodiments, the interpolation manager constructs each of the multiple interpolated frames between the two existing frames by selecting a pixel set from each existing frame, generating associated meshes, estimating flow motions in each direction, generating corresponding motion compensated meshes, computing warped images corresponding to the existing frames and combining the warped images into an interpolated frame.

[0005] For each constructed interpolated frame, the interpolation manager uses different classification criteria to select the pixel sets, such that the pixel sets selected for each of the motion compensated interpolated frames vary, and hence the interpolated frames vary as well. The interpolation manager fuses the multiple interpolated frames into a single, output interpolated frame. The interpolation manager combines each of the pixels in the multiple interpolated frames to give a final pixel output color in the single, output interpolated frame. In some embodiments, the interpolation manager achieves this by using a voting methodology, and assuming that all votes are equally accurate.

[0006] The features and advantages described in this summary and the following detailed description are not all-inclusive, and particularly, many additional features and advantages will be apparent to one of ordinary skill in the art in view of the drawings, specification, and claims hereof. Moreover, it should be noted that the language used in

the specification has been principally selected for readability and instructional purposes, and may not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter.

Brief Description of the Drawings

[0007] Figure 1 is a block diagram, illustrating an interpolation manager constructing three interpolated frames between two existing frames, according to some embodiments of the present invention.

[0008] Figure 2 is a flowchart, illustrating steps for an interpolation manager to construct interpolated frames, according to some embodiments of the present invention.

[0009] Figure 3 is a block diagram, illustrating an interpolation manager fusing three interpolated frames into a single, output interpolated frame, according to some embodiments of the present invention.

[0010] Figure 4 is a flowchart, illustrating steps for an interpolation manager to fuse multiple interpolated frames into a single, output interpolated frame, according to some embodiments of the present invention.

[0011] The figures depict embodiments of the present invention for purposes of illustration only. One skilled in the art will readily recognize from the following discussion that alternative embodiments of the structures and methods illustrated herein may be employed without departing from the principles of the invention described herein.

Detailed Description

[0012] Figure 1 illustrates a high level overview of a system 100 for performing some embodiments of the present invention. As illustrated in Figure 1, an interpolation manager 101 constructs multiple motion compensated interpolated frames 103 between two existing frames 105. It is to be understood that although the interpolation manager 101 is illustrated as a single entity, as the term is used herein an interpolation manager 101 refers to a collection of functionalities which can be implemented as software, hardware, firmware or any combination of the three. Where an interpolation manager 101 is implemented as software, it can be implemented as a standalone program, but can also be implemented in other ways, for example as part of a larger program, as a plurality of separate programs, or as one or more statically or dynamically linked libraries.

[0013] Figure 1 illustrates the interpolation manager 101 constructing 3 interpolated frames 103 between existing frames 1 and 2. Of course, 3 is only an example of a number of interpolated frames 103 that the interpolation manager 101 can construct. It is to be understood that the interpolation manager 101 can construct any integer N interpolated frames 105, wherein N is greater than 1. The specific number of interpolated frames 103 to construct in a given embodiment is a design parameter.

[0014] As explained in detail below, the interpolation manager 101 builds each interpolated frame 103 such that the multiple interpolated frames 103 vary in their internal motion compensation between the two existing frames 105. This variance is used to robustly fuse the multiple motion compensated interpolated frames 103 into a single, output interpolated frame 301, as explained in detail below.

[0015] Figure 2 illustrates steps for the interpolation manager 101 to construct the multiple interpolated frames 103, according to some embodiments of the present invention. For each interpolated frame 103 to build, the interpolation manager 101 selects 201 a first set of pixels in the first existing frame 105, and a corresponding second set of pixels in the second existing frame 105.

[0016] In some embodiments, each pixel set is selected 201 by classifying some pixels in the corresponding existing frame 105 as having a high spatial frequency contents, and selecting the classified pixels. Because not all parts of an image contain complete motion information, it is beneficial to select only those pixels in the existing frames 105 with high spatial frequency contents. This criterion can be defined so that the motion estimation step described below is well-suited to apply to the selected pixels. In such embodiments, the specific classification criteria used is different for each of the N motion compensated interpolated frames 103, such that the pixel sets selected for each of the N motion compensated frames vary.

[0017] Next, the interpolation manager generates 203 a first mesh for the first set of pixels and a second mesh for the second set of pixels. In one embodiment, this involves the interpolation manager 101 fitting a polygonal mesh to each set of pixels. This can be achieved, for example, by using a constrained Delaunay triangulation over each set of selected pixels, using the edges present in the associated existing frame 105 as imposed boundaries (i.e., the interpolation manager forces the triangulation to conform to those fixed edges).

[0018] After generating 203 the meshes, the interpolation manager 101 estimates 205 a first flow of motion from the first set of pixels to the second set of pixels, and a

second flow of motion from the second set of pixels to the first set of pixels. In one embodiment, the interpolation manager 101 estimates 205 these flows of motion by using the classical optical flow constraint equation: $I_x * u + I_y * v + I_t = 0$, where (u, v) are the unknown components of the flow and x, y and t stand for differentiation.

[0019] In some embodiments, during motion estimation the interpolation manager 101 exploits information from a small neighborhood around an examined pixel, by collecting constraints of neighboring points and solving the resulting over-constrained set of linear equations. Because the set of pixels selected 201 for motion estimation is optimized, the invention can avoid the known ill-conditioning drawback of classical local motion estimation techniques.

[0020] The interpolation manager 101 then generates 207 a first motion compensated mesh corresponding to the first existing frame 105, and a second motion compensated mesh corresponding to the first existing frame 105. To generate a motion compensated mesh corresponding to an existing frame 105, the interpolation manager 101 uses the corresponding polygonal mesh, and the corresponding motion estimation described above. Let P_1 denote a point in a polygonal mesh associated with a source existing frame 105, and P'_1 its correspondence in the destination existing frame 105 obtained during the motion estimation step. For each point P_1 in the polygonal mesh, the interpolation manager 101 computes a corresponding point P in the new mesh by moving the point P_1 to a new position P , which can be calculated as $P = (1.0 - t) * P_1 + t * P'_1$.

[0021] The interpolation manager 101 computes 209 two warped images, each corresponding to one of the existing frames 105. The interpolation manager 101 computes 209 a warped image for an existing frame by warping the frame using the

corresponding polygonal and motion compensated meshes. The interpolation manager 101 then combines 211 the two warped images ($W1$, $W2$) by linearly mixing them to produce a single interpolated frame 103 M_i . For example, the linear mixing of the warped images can take the form: $M_i = (1.0 - t) * W1 + t * W2$.

[0022] The interpolation manager 101 repeats steps 201-209 N times, using N different pixel classification criterion, thereby producing N different interpolated frames 103.

[0023] Figure 3 illustrates the interpolation manager 101 fusing 3 interpolated frames 103 into a single, output interpolated frame 301 between the two existing frames 105 (not illustrated in Figure 3). In this process, each of the pixels in the multiple interpolated frames 103 are combined together to give a final pixel output color in O , the single, output interpolated frame 301. In some embodiments, the interpolation manager 101 achieves this by using a voting methodology, and assuming that all votes are equally accurate. For example, turning to Figure 4, for each given pixel $P(x,y)$ in O , the interpolation manager 101 can apply 401 a scalar median filter componentwise to the corresponding pixels $P_i(x,y)$ in each interpolated frame 103, thus producing a pixel $P_m(x,y)$. The interpolation manager 101 then sets 403 pixel $P(x,y)$ in O to the pixel $P_i(x,y)$ that is color nearest (in the sense of $L1$ norm) to $P_m(x,y)$. The interpolation manager 101 can employ these steps to determine every pixel in the single, output interpolated frame 301. This solution is the output color of $P(x,y)$.

[0024] Using this new fusion strategy to combine the results of different motion compensated interpolated frames 103 generates results of a higher quality for the given problem than could be achieved by any single technique alone. The use of this new

fusion strategy produces smooth motion compensated interpolation, particularly robust to the presence of occlusion.

[0025] As will be understood by those familiar with the art, the invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. Likewise, the particular naming and division of the modules, features, attributes, methodologies, managers and other aspects are not mandatory or significant, and the mechanisms that implement the invention or its features may have different names, divisions and/or formats. Furthermore, as will be apparent to one of ordinary skill in the relevant art, the modules, features, attributes, methodologies, managers and other aspects of the invention can be implemented as software, hardware, firmware or any combination of the three. Of course, wherever a component of the present invention is implemented as software, the component can be implemented as a standalone program, as part of a larger program, as a plurality of separate programs, as a statically or dynamically linked library, as a kernel loadable module, as a device driver, and/or in every and any other way known now or in the future to those of skill in the art of computer programming. Additionally, the present invention is in no way limited to implementation in any specific programming language, or for any specific operating system or environment. Accordingly, the disclosure of the present invention is intended to be illustrative, but not limiting, of the scope of the invention, which is set forth in the following claims.